

Available online at www.sciencedirect.com**SciVerse ScienceDirect**

Procedia Engineering 15 (2011) 5541 – 5546

**Procedia
Engineering**www.elsevier.com/locate/procedia**Advanced in Control Engineering and Information Science**

Traffic Congestion Detection Based On GPS Floating-Car Data

ZHANG Yong-chuan^a, ZUO Xiao-qing^{a*}, ZHANG li-ting^a, CHEN Zhen-ting^{a,b}*a. Faculty of Land Resource Engineering, Kunming University of Science and Technology, Kunming 650093, China;**b. Department of Computer Information, Kunming Metallurgy College, Kunming 650033, China*

Abstract

Congestion has been a major challenge to urban traffic system. Based on GPS floating-car data (FCD), a method to acquire traffic congestion information is studied. And then, using about 6,000,000 pieces of data generated by 500 taxis in one day as experiment data, we extract the traffic congestion information of a big city in China through GPS data preprocessing, map matching, travel speed estimation, and several other key steps, and finally a map which exhibit the traffic congestion distribution of the city is produced.

© 2011 Published by Elsevier Ltd. Open access under [CC BY-NC-ND license](#).

Selection and/or peer-review under responsibility of [CEIS 2011]

Key words: GPS floating-car data (FCD); traffic congestion; map matching; link travel speed

1. Introduction

Controlling distribution information of traffic congestion is extremely important in traffic management, so, how to easily get this information has come to an issue. With the roaring growth of cars carrying GPS, using GPS floating car technology to collect traffic information becomes more and more cost-effective, so we can use it to collect traffic information in a large-scale. Compare with the traditional fixed detection devices (inductive loops, video cameras and radar based sensors, etc.), the floating car data technique can collect traffic information more accurate, more comprehensive, in a larger scale and in real-time, so we consider using GPS floating car data to extract traffic congestion information. Since the early 1980's, a large number of researches and experiments have been carried out to study the feasibility of it both home

*Corresponding author. ZUO Xiao-qing
E-mail address: ieee2010@foxmail.com

and abroad [1]. For example, the ADVANCE projects of United States [2, 3, 4], the FVD system of Britain, the DDG system of Germany, the IP Car system of Japan [5], etc. However, they mainly focused on the estimation of link travel time, and link travel speed, and did not concentrate in traffic congestion information that much. This paper proposed a method of using the GPS floating car data to detect the urban traffic congestion information, and has done some experiments to test the feasibility.

The remainder of this paper is structured as follows. Section 2 introduces some key method and algorithm of congestion detection, and experiments of using 6,000,000 pieces of GPS floating car data with our algorithm to detect the traffic congestion information of a major city in China is reviewed in Section 3. Section 4 is conclusion.

2. Theories and Methods

Using GPS floating car data to detect urban traffic congestion information includes several steps: data preprocessing, map matching, travel speed estimation and traffic congestion classification. The related methods are discussed as follows.

2.1. Data and data pre-processing

FCD is generally updated at intervals ranged from 1 second to 10 minutes in NMEA-0183 mode and is delivered using GPRS sentence. After interpreted and processed, the data includes information about the ID, Time, Longitude, Latitude, Speed, Orientation and others. Table 1 makes detailed description of it [6].

Table 1. Description of the GPS FCD

Field	Example	Description
ID	221213582622	DI of taxi ,unique
Time	2011-3-22 13:20:15	Beijing Time
Longitude	102.763948	
Latitude	25.091095	
Speed	47	Units: km/h
Orientation	102	Units: degree

Usually, the FCD is not good enough to be used directly. The errors and corresponding amending methods are as follows:

- Part of the location-data's error is too large to fix the estimating position to corresponding roads. They are abnormal points (outlier) which should be removed.
- When under bridges, in tunnels, among tall and dense buildings and in other areas, GPS receivers can't receive the satellite signal, which lead to the data missing. We interpolate the missing points according to the vehicles' historical trajectory.
- The speeds of GPS floating car which are far greater or smaller than most of the speed on the same road at the same time. We drop those points, and interpolate it according to the speeds presented by points around.
- The position data is given using the WGS84 coordinate system and the road network is often presented using other coordinate system. We unified them through coordinate transformation.

2.2. Map Matching

For the reason that the positioning accuracy of GPS data is limited, vehicles can't be located on the road accurately in most of time (as known to all, vehicles should be on the road). If we want to extract travel speed of the road, we should firstly "pull" the inaccurate location points (we also call it estimated position) to road, and as a matter of fact, this process is called Map Matching [7]. In the traffic transportation area, the most commonly used map-matching algorithms are the algorithm of point-to-curve and the algorithm of curve-to-curve. In this paper, we use the map matching algorithm that combines the point-to-curve with the curve-to-curve to improve the match efficiency.

The algorithm of point-to-curve, which is considered to be uncomplicated, is suitable for situations that the road network waited to be mapped is simple, and it uses projection distance and angle between road and direction of the estimated point as constraint. Figure 1 (a) shows that estimated points P and P' are projected to the adjacent roads L_1 , L_2 individually, and the projection points are P_1 , P_2 and P'_1 , P'_2 . According to the knowledge of analytic geometry, the distance between point and line can be calculated as

$$d = |ax + by + c| / \sqrt{a^2 + b^2} \quad (1)$$

So, we can get d_1 , d_2 , d'_1 , d'_2 which represent the distance from P to the L_1 , L_2 and P' to the L_1 , L_2 respectively, and then according to the rule of nearest projection distance constraint $P_{matched} = \min(d_i)$, we can determine the point P match to point P_1 on L_2 , P' match to the point P'_1 on L_1 , and drop the point P_2 and P'_2 .

In the case of the road crossing, the algorithm which considers both distance and angle constraint could be more accurate. As shown in Figure 1(b), at the road crossing, should point P matched to the P_1 or P_2 ? According to the matching rule

$$P_{matched} = \min(k_1 d_i + k_2 \Delta \theta_i) \quad (2)$$

It considered distance and angle constraint, and $\Delta \theta_i$ is the angle between driving direction and the road direction, k_1 , k_2 is the weight coefficient of the distance and the angle respectively, and meet with the equation $k_1 + k_2 = 1$, generally speaking, $k_1 = 0.5$, $k_2 = 0.5$. Through the rule (2), we can make the conclusion that the point P should be matched to P_1 on L_1 , drop P_2 .

The algorithm of curve to curve, which is considered to be a little complicated, is suitable for situations that the road network waited to be mapped is complex. The algorithm is a little complex but accurate in most of the time, and the matching rule is that the deviation between the trajectory of vehicle and the matching segment of road is smallest. As it is shown in Figure 2, the road segments $1 \rightarrow 2 \rightarrow 3$ is a travel trajectory, AB and EB are road sections needed matching.

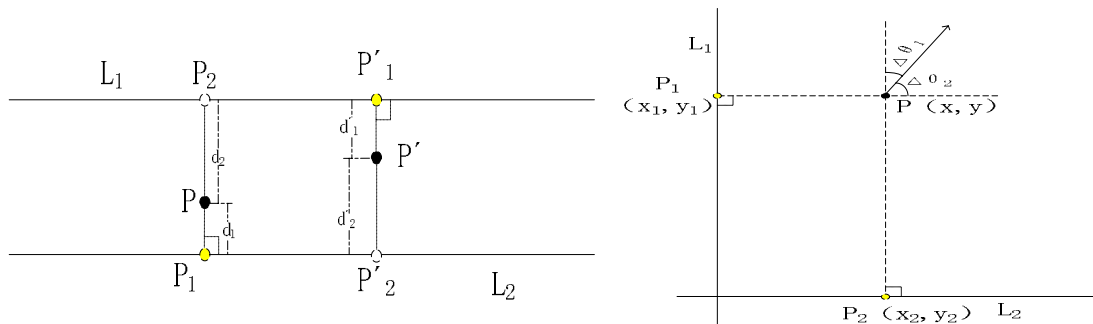


Fig.1. (a) Point to curve map matching algorithm (only consider the constraint of distance); (b) Map matching algorithm considered the constraints of distance and angle

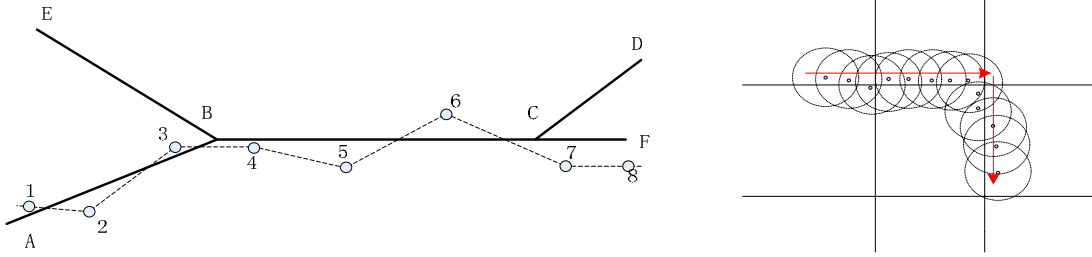


Fig.2. (the left) the algorithm of curve to curve map matching; Fig.3 (the right) shows the schematic diagram of searching for the matched road section

Given that a road section L_i which need matching is the function of the time t , $L_i = a(t)$, and a travel trajectory L is also the function of the time t , $L = L = b(t)$, then the distance between the two is

$$d_i = \|L_i - L\| = \int_{t_i}^{t_{i+1}} |a(t) - b(t)| dt \quad (3)$$

Then the rule can be described as:

$$L_{matched} = \min(d_i) \quad (4)$$

According to this rule we can match the road segments $1 \rightarrow 2 \rightarrow 3$ to AB, while abandoning EB.

In addition, if we want to achieve large amounts of data's map matching, the matching efficiency is a challenge which needs to consider. In every matching process of a point, taking the entire road segment as the selecting road need matching is a method without efficiency, in fact, it is not necessary to consider every road as matching road, too. We can calculate the maximum error radius according to the positioning errors and the maximum travel distance at one interval.

$$R = \max(\Delta T * v) \quad (5)$$

In equation (5), v is instantaneous speed, ΔT is the interval. The matching points are only possible in the circle whose center is the estimate point, and the radius is R . So, using the circle as searching area, only the roads within or intersected with the circle can be selected as the road to be matched, thus, the matching efficiency is improved. And when the matching segment number $i=1$, we use the algorithm of point to curve to improve the matching speed; when the matching segment number $i \geq 2$, we regard the road network is complex, we use the algorithm of curve to curve to improve the matching accuracy. Figure 3 shows the progress of map matching along the direction of the red arrow.

2.3. Extracting the travel speed

After map matching, the floating car location data are corrected to the road, so we can rebuild the trajectory. According to the two continuous location points' distance l_j on the road and the interval ΔT between them, we can calculate the average speed $\bar{v}_j = l_j / \Delta t$. Thus, the average travel speed of one road is

$$\bar{v}_i = (\sum_0^m v_j) / m \quad (6)$$

In equation (6), road index number is i and contains $m+1$ point. Then average the travel speed of n cars at the same time, we can get the average link travel speed on every interval.

3. Experiments

The data for experiment are from 500 taxis which carrying GPS receivers, totaling about 600 million data in one day, and the interval of data sampling is 15 seconds. After the data preprocessing, we can get qualified data, then carries on map matching. Figure 4 shows the process and result of map matching. Among them, figure4 (a) shows the state of GPS floating car data which is projected on the road net after data preprocessing, and the red point shows the instantaneous position in space of a GPS floating car. As can be seen from the figure, most floating cars are not on the road. We can "pull back" these points to the road network through the map matching technique. Figure 4(b) demonstrates the effect of a common matching algorithm (point-to-line), and the red points represent projection position, and the yellow points represent matched position. As can be seen from the graph the most points can exactly match to network using then common algorithm, but in road intersections where road network is a little complex, the matching is fail, like the red circle 1, 2, 3 showed in the figure. Figure 4(c) shows the effect of map matching algorithm of point-to-line combined with line-to-line, contrast with figure 4 (b), projection points in red circle 1, 2, 3 can match on road network correctly.



Fig.4. (a) The first picture shows FCD project on road map; (b) the second picture shows the effect of common map matching algorithms; (c) the third picture shows the effect of our map matching algorithm



Fig. 5. shows the trajectory reconstruction of FCD

We can rebuild the trajectory of the floating car via connecting the matched points on the road according to the time sequence after completing map matching. Figure 5 shows the effect of floating car's trajectory reconstruction, yellow points for matching position, red line for floating car driving trajectory, blue arrows say driving direction. Further, we can calculate the travel speed after extracting the length and time passed of floating car on a road.

We count out average travel speed of 2,000 road in X city, statistic them from small to large. The maximum speed is 100km/h in all roads and the road travel speed over 50km/h is in the minority and the speed below 50km/h are well-distributed. To take the travel speed of each road as the foundation, and referring to the "Index System of Traffic Management Evaluation of Urban Road "(the 2008 edition), the urban road network can be divided into 5 categories. They are serious congestion (≤ 15 km/h), congestion

(<=25km/h), normal (<= 35km/h), smooth (<= 50km/h) and fast (> 50km/h) separately. And figure 6(a) is the road map in X city, after classify the city roads into categories via methods above, then the road can be color-coded according to the FCD travel speed categories as shown in figure6(b). As can be seen from the graph, in the area of city center, traffic is congestion, second-ring traffic is ordinary, third-ring traffic is unobstructed, and this case is accordant with our usual actual experience.

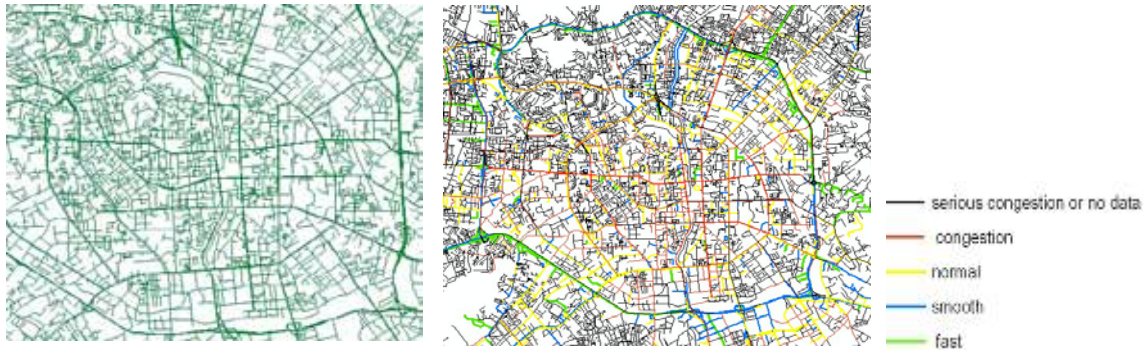


Fig. 6. (a) Road map network in a city (part); (b) Distribution of the congestion

4. Conclusions

In this paper a method to acquire traffic congestion information using floating car data technique is studied, including main techniques of data cleaning, coordinate transformation, map matching, trajectory reconstruction, segment travel speed estimation and road congestion classification. And we mainly explored map matching algorithms of points-to-line combined with line-to-line, and methods of narrowing the search radius in map matching to improve the efficiency of the map matching. Based on these techniques and methods, we detected the information of traffic congestion status in X city in one day. Research results in this paper have reference significance to solve urban traffic congestion problem, and to formulate rational traffic counseling scheme.

5. Acknowledgments

This paper is sponsored by National Nature Science Foundation of China (41061043).

6. References

- [1] http://www.tranbbs.com/Techarticle/ITS/Techarticle_43308.shtml.
- [2] David E. Boyce, Allan Kirson, Joseph L. Schofer. Design and Implementation of ADVANCE: The Illinois Dynamic Navigation and Route Guidance Demonstration Program.
- [3] Christopher L. Saricks, etc. Formal Evaluation of the ADVANCE Targeted Deployment.
- [4] BOOZ, ALLEN & HAMILTON, Federal Highway Administration Intelligent Transportation Systems Compendium of Field Operational Test Executive Summaries.
- [5] SUNAHARA HIDEKI, SATO MASAOKI, UEHARA KEISUKE. IPCar: Building the Probe Car System with the Internet. IEICE Transactions on Communications (Japanese Edition), VOL.J85-B; NO.4; PAGE.431-437(2002).
- [6] <http://www.cnblogs.com/anhlyiling/archive/2008/12/04/1347657.html> Christopher
- [7] E. White, David Bernstein, Alain L. Kornhauser, Some map matching algorithms for personal navigation assistants, Transportation Research Part C 8 (2000) 91-108.